PROYECTO FINAL

INTEGRANTES

Arnaldo Benavides Rodriguez

Donny Escolar Gaviria

Nicolas Poveda Alvarado

Santiago Poveda De La Hoz

Michael Valero Polo

DOCENTE

Elias Ruiz Niño

UNIVERSIDAD DEL NORTE

DIVISIÓN DE INGENIERÍAS

BARRANQUILLA

2021

Indice

Introducción al tópico.	**2**
Justificación del estudio.	**2**
Objetivo general.	**3**
Objetivos específicos.	**3**
Análisis exploratorio de datos.	**3**
Formulación del problema.	**6**
Pasos solución del problema	**7**
Contratiempos.	**12**
Conclusiones.	**13**
Referencias	**13**

1. Introducción al tópico.

Los coronavirus (CoV) son virus que surgen periódicamente en diferentes áreas del mundo y que causan Infección Respiratoria Aguda (IRA), es decir gripa, que pueden llegar a ser leve, moderada o grave. El nuevo Coronavirus (COVID-19) fue catalogado por la Organización Mundial de la Salud como una emergencia en salud pública de importancia internacional (ESPII), pues la rápida transmisión del mismo y su alta tasa de letalidad tomó por sorpresa a gran parte de la población; El 6 de marzo de 2020 se confirmó el primer caso positivo en Colombia.

El Ministerio de Salud y Protección Social a través del aplicativo SegCovid recibe diariamente de las IPS la ocupación UCI, estos datos acumulados se publican posteriormente en la página web y son una fuente importante que nos permite medir el estado del país. Más adelante, para análisis, investigaciones y fines complementarios, el Ministerio entrega al INS (Instituto Nacional de Salud) información nominal que recopila datos de hasta dos semanas atrás. Por eso, esta información jamás debe ser utilizada para estimar ocupación actual o reciente de las unidades de cuidados intensivos.

Los modelos de machine learning permiten que analistas e investigadores entrenen conjuntos de datos para mejorar las asociaciones entre n variables seleccionadas con el fin de poder generar predicciones más acertadas y distinguir patrones que pueden escaparse de la observación humana. Actualmente existen diversos tipos de modelos predictivos enfocados a datasets, por ello, es pertinente estudiar y comparar a fondo cuál de ellos se acopla mejor a nuestros datos.

2. Justificación del estudio.

Es necesario analizar e interpretar los datos que se obtuvieron a comienzos del 2020 para poder tomar acciones acertadas ante la actual problemática sanitaria que representa el brote del virus Covid-19 y sus variantes. El analizar rangos de edad, sexo, lugar de residencia y

demás varibles, nos permitirá plantear hipotesis y hacer predicciones que podrían ayudarnos a comprender de una mejor forma el comportamiento del virus.

3. Objetivo general.

Como propósito principal de este proyecto nos planteamos procesar y analizar los datos de los primeros mil casos positivos de covid-19 reportados en el país. De tal manera que podamos conocer la forma en que distintos factores contribuyeron a la expansión del virus por todo el territorio y comparar los scores arrojados por los modelos de machine learning propuestos; esto con el propósito de encontrar cual se acoge mejor a nuestro conjunto de datos.

4. Objetivos específicos.

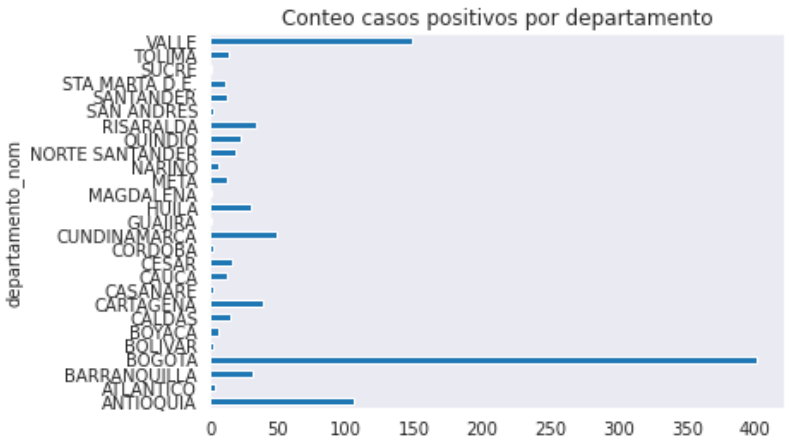
- Ajustar los datos a diversos modelos de machine learning y con ellos determinar el grado de relación entre las variables seleccionadas.
- Ejecutar predicciones y hallar el porcentaje de coincidencia.
- Graficar y explicar las relaciones entre variables de interés en el dataset.

5. Análisis exploratorio de datos.

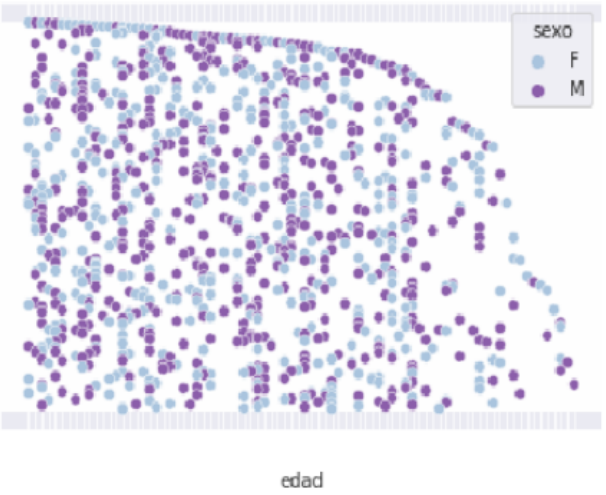
En el análisis exploratorio vimos que el set de datos importado posee 23 variables (Columnas) y 1000 observaciones (Filas), lo cual nos permitió de manera inicial tener una capacidad de análisis moderada. Usando la función info() de la librería **Pandas** se observó que todos los datos son de tipo objeto; esto facilitó el manejo de los mismos de una forma rápida y sencilla.

#	Column	Non-Null Count	Dtype
0	fecha_reporte_web	1000 non-null	object
1	id_de_caso	1000 non-null	object
2	fecha_de_notificaci_n	1000 non-null	object
3	departamento	1000 non-null	object
4	departamento_nom	1000 non-null	object
5	ciudad_municipio	1000 non-null	object
6	ciudad_municipio_nom	1000 non-null	object
7	edad	1000 non-null	object
8	unidad_medida	1000 non-null	object
9	sexo	1000 non-null	object
10	fuente_tipo_contagio	1000 non-null	object
11	ubicacion	1000 non-null	object
12	estado	1000 non-null	object
13	pais_viajo_1_cod	477 non-null	object
14	pais_viajo_1_nom	477 non-null	object
15	recuperado	1000 non-null	object
16	fecha_inicio_sintomas	956 non-null	object
17	fecha_diagnostico	1000 non-null	object
18	fecha_recuperado	950 non-null	object
19	tipo_recuperacion	950 non-null	object
20	per_etn_	1000 non-null	object
21	nom_grupo_	14 non-null	object
22	fecha_muerte	52 non-null	object

Observando nuestras gráficas más representativas en función de las variables escogidas, vimos que en el conteo de casos por departamento destacaron: **Valle**, **Antioquia** y **Cundinamarca** (Bogotá DC), esto al ser los más poblados del país; ahora sabemos que esta sería una tendencia que se mantendría por varios meses. También, ciudades importantes como Barranquilla y Cartagena tuvieron una cantidad considerable de casos positivos si tenemos en cuenta que en ellas hay una menor población con respecto a los anteriores. Es decir, podemos inferir que los polos turísticos del país representaron un foco de contagio al ser los primeros en tener un contacto con personas infectadas que provenían del estrangero.

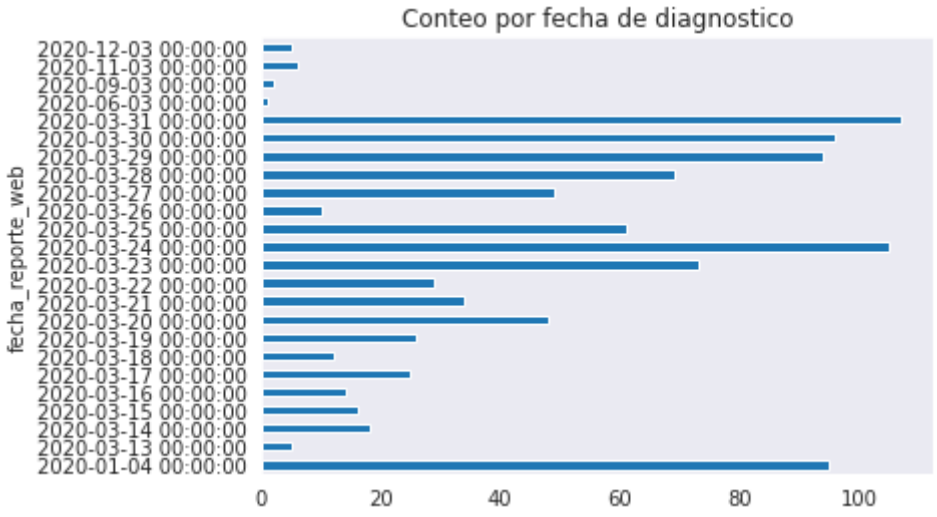


Observando lo anterior, se volvió importante determinar qué rango de edad fue el más vulnerable según el conteo por dicha variable. De ello pudimos notar que a mayor edad el número de casos disminuyó, esto debe ser una cuestión a tener en cuenta ya que según estudios realizados a edades más tempranas existe un alta probabilidad de viajar, lo que nos dice que este es un factor importante que influye en el esparcimiento del virus.

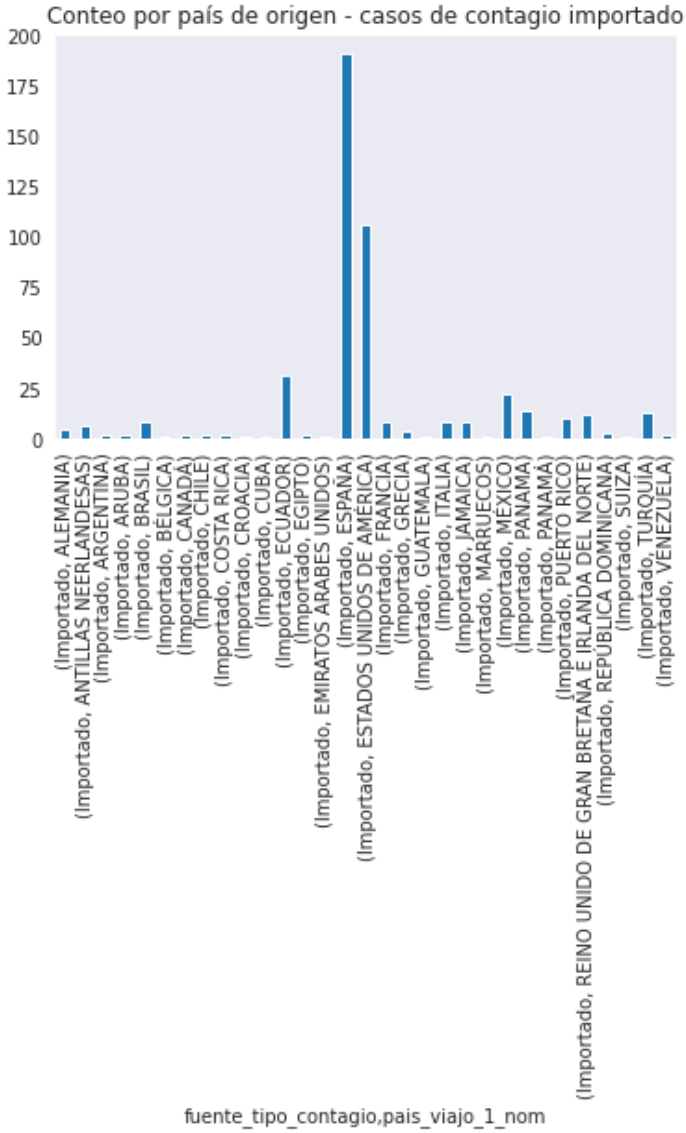


Se infiere entonces que el mayor número de contagio se dió entre la población de adultos jóvenes, además, el número de contagios disminuyó mientras se avanzó en la escala de edad.

Por otra parte, se graficó el número de casos vs fecha reporte web y se observó una tendencia al aumento con el paso de los días (6 de marzo 2020 - 4 de abril 2020).



Por último, una gráfica del conteo de aquellas personas que adquirieron el virus en el extrangero nos dice que el país con mayor número casos fue **España**, seguido de la potencia mundial **Estados Unidos**.



6. Formulación del problema.

Actualmente, el mundo se enfrenta a una de las mayores crisis sanitarias de los últimos tiempos. El coronavirus COVID-19 está mutando y revelando variantes que amenazan la estabilidad de nuestra especie, es por ello que estudios como el nuestro representan la esperanza que tienen millones de personas de poder salir de esta problemática.

Teniendo en cuenta el dataset seleccionado: https://www.datos.gov.co/Salud-y-Protecci-n-Social/Casos-positivos-de-COVID-19-en-Colombia/gt2j-8ykr/data. Pensamos que sería interesante realizar un modelo de machine learning enfocado a las variables: “sexo”, “departamento”, “edad” y “ciudad_municipio”; y mediante ellas, realizar predicciones. Sin embargo, encontrar un modelo apropiado representa un reto en sí pues nos enfrentaremos a una variable categórica y otras numéricas. En este proyecto trataremos de encontrar una solución a esto.

7. Pasos solución del problema

- **Librerias**

- Utilizamos la librería de python **pandas**, la cuál es una biblioteca para manipulación y análisis de datos. En particular, ofrece estructuras de datos y operaciones para manipular tablas numéricas y series temporales. En este proyecto la usamos para implementar funciones en el dataset obtenido de la web.

- La biblioteca UrlLib es utilizada en este proyecto para obtener los datos mediante una dirección URL API.

- Para la creación de gráficas y visualización de datos es usada la librería Seaborn la cuál se basa en matplotlib y nos permite comprender la distribución en el dataset.

- En machine learning y los respectivos modelos utilizamos la librería Scikit-learn que se usa para análisis de datos predictivo y se basa en la librería numérica NumPy.

- **Linear regression:**

  Aquí el set se divide y se toma la variable dependiente “ciudad_municipio” en función del “departamento” y de la “edad” para ver si un modelo de regresion lineal puede explicar la dinamica de los datos.

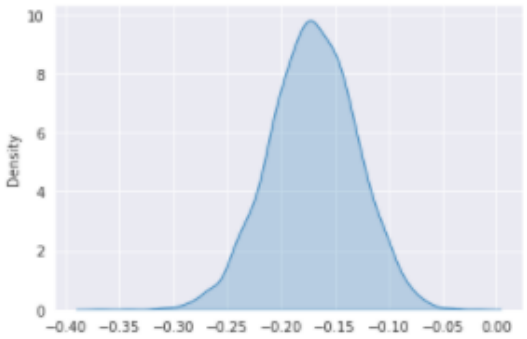
```
[ ] y = df_covid[['ciudad_municipio']].values.astype(np.float32)
    X = df_covid[['departamento','edad']].values.astype(np.float32);

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=10)
    lr = LinearRegression(fit_intercept=False);

    #We train with _train
    lr_fit = lr.fit(X_train, y_train);

    #We validate with _test
    lr_score = lr_fit.score();

    print(f'* Score of linear model for covid data set {lr_score}');
```

```
* Score of linear model for covid data set -0.15975735032645044
```

```
[ ] N = 5000;
    scores = [];
    alphas = [];
    for i in range(0, N):
      X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                          test_size=0.3,
                                                          random_state=i+4);
      lr = LinearRegression(fit_intercept=False);
      lr_fit = lr.fit(X_train, y_train);
      lr_score = lr_fit.score(X_test, y_test);
      scores.append(lr_score);
      alphas.append(lr_fit.coef_[0]);
    scores = np.array(scores, dtype=np.float32);
    alphas = np.array(alphas, dtype=np.float32);
```

```
[ ] sns.kdeplot(scores, shade=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc7fe539a10>
```



Sin embargo, observamos que este modelo arroja un score de -0.15975735032645044, lo cual no es nada bueno pues indica que los datos no explican su objetivo en absoluto; con la función .score() halla el coeficiente de determinación R2 de forma predeterminada y se utiliza para evaluar el desempeño.

- **Logistic model:**

Con el modelo de regresion logistico tenemos que su analisis nos permite precedir el comportamineto de una variable categorica en funcion de variables periodicas, en este caso intentamos cambiar de perspectiva y construimos un modelo de “sexo” en funcion de “edad” y “departamento”, note que el “sexo” es una variable categorica. Esto corresponde a un problema de clasificacion.

$$\text{Sexo} = \frac{1}{1 + \exp(-(a_1 \cdot \text{Departamento} + a_2 \cdot \text{Edad} + a_0))} + \epsilon$$

```
from sklearn.linear_model import LogisticRegression
datos = df_covid.values
y_logi = datos[:,-14];
X_logi = df_covid[['departamento','edad']].values.astype(np.float32)
X_tra, X_test, y_tra, y_test = train_test_split(X_logi, y_logi, test_size=0.3, random_state=10);
```

1. Train the model

```
[16] lr = LogisticRegression(solver='liblinear');
     lr_fit = lr.fit(X_tra, y_tra);
```

2. Validation

```
lr_fit.score(X_test,y_test)

0.5
```

Como se puede observar, se obtuvo un score del 50%; algo más acorde a nuestras expectativas. Luego, realizamos una predición con base en la variable “sexo” y se obtuvieron los siguientes reultados.

```
ma_fe = lr_fit.predict(X_test)
ma_fe
```

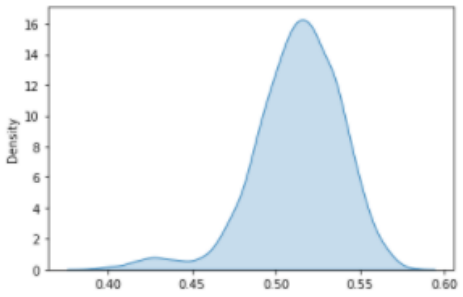
```
array(['M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'F', 'M',
       'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M',
       'M', 'M', 'M', 'M', 'M', 'M', 'F', 'M', 'M', 'M', 'M', 'M', 'M',
       'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M',
       'M', 'M', 'M', 'M', 'M', 'M', 'F', 'M', 'M', 'M', 'M', 'M', 'F',
       'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M',
       'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M',
       'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M',
       'M', 'F', 'M', 'M', 'M', 'M', 'F', 'M', 'M', 'M', 'M', 'M', 'M',
       'M', 'M', 'M', 'M', 'M', 'F', 'F', 'M', 'F', 'M', 'M', 'F', 'M',
       'F', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'F', 'M', 'M', 'M', 'M',
       'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M',
       'M', 'M', 'M', 'M', 'M', 'F', 'M', 'M', 'F', 'M', 'M', 'M', 'M',
       'M', 'M', 'M', 'M', 'M', 'M', 'M', 'F', 'M', 'M', 'M', 'M', 'M',
       'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M',
       'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M',
       'M', 'M', 'F', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'F', 'M', 'M',
       'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'F',
       'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'F',
       'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M',
       'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M',
       'M'], dtype=object)
```

A simple vista podemos concluir que el genero masculino lidera la predicción.

```
N = 5000;
scores = [];
alphas = [];
for i in range(0, N):
  X_tra, X_test, y_tra, y_test = train_test_split(X_logi, y_logi, test_size=0.3, random_state=i+4);
  lr = LogisticRegression(solver='liblinear',fit_intercept=False);
  lr_fit = lr.fit(X_tra, y_tra);
  lr_score = lr_fit.score(X_test, y_test);
  scores.append(lr_score);
  alphas.append(lr_fit.coef_[0]);
scores = np.array(scores, dtype=np.float32);
alphas = np.array(alphas, dtype=np.float32);
```

```
sns.kdeplot(scores, shade=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff4a66e9650>
```



Se puede ver como la probabilidad de la varible categorica dependiente sexo como funcion de la edad y el departamento tiene un score de tendencia de un poco más 50% de las veces en este modelo (n=5000).

- **Naive Bayes:**

Dado que un modelo basado en problemas de clasificasion describió mejor nuestros datos decidimos probar ahora el algoritmo de Nalve Bayes con el propósito de obtener una clasificación probabilística.

```
from sklearn.preprocessing import StandardScaler
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import confusion_matrix,accuracy_score
```

Partimos nuestros datos y realizaremos un escalado de características al conjunto de entrenamiento y prueba.

```
y_NB = df_covid[['ciudad_municipio']].values.astype(np.float32)
X_NB = df_covid[['departamento','edad']].values.astype(np.float32);
X_train, X_test, y_train, y_test = train_test_split(X_NB, y_NB, test_size=0.3, random_state=10)
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

Entrenamiento del modelo **Naive Bayes** en el set de entrenamiento

```
classifier = GaussianNB()
classifier.fit(X_train, y_train)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:985: DataConversionWarning: A column-vector
  y = column_or_1d(y, warn=True)
GaussianNB()
```

Predicción

```
y_pred  =  classifier.predict(X_test)
y_pred
```

```
array([25754., 76001., 76001., 11001., 11001., 76001., 19001., 76001.,
       41001., 19001., 41001., 47001., 76001., 41001., 11001., 11001.,
       50001., 20001., 11001., 76001., 11001., 63001., 11001., 17001.,
        5376., 11001., 73001., 19001., 11001., 11001., 11001., 11001.,
       13001.,  5001., 11001., 73001., 76001., 73001., 11001., 41001.,
       25214., 52001., 76001., 11001., 11001., 11001., 73001., 11001.,
       25473., 20001., 66001., 11001.,  5001., 25817., 25769., 76001.,
       11001.,  5001.,  8001., 11001., 11001., 41001., 11001., 76001.,
       13001., 76001., 11001., 11001., 76001., 11001., 11001., 19001.,
       11001., 76001.,  5001., 11001., 11001., 50001., 19001., 11001.,
       11001.,  5001.,  8758., 11001., 66001., 25214., 11001., 11001.,
       11001., 17873., 11001., 11001., 11001., 54001., 11001., 54001.,
       76001., 66001., 50001., 11001.,  5001., 11001., 41001., 11001.,
        5001., 76001., 13001., 11001., 11001., 68276., 11001., 13001.,
       63001., 54001., 25214., 66001., 11001., 41001.,  8001., 41001.,
        5001., 76001., 76001., 13001., 25754., 63001., 11001., 25214.,
       25286., 11001., 41001., 11001., 76001.,  5001., 25754.,  8001.,
        8001., 52001., 13001., 11001.,  5001., 13001., 66001., 13001.,
       25214., 76001.,  5001., 76001., 11001., 76001., 20001., 13001.,
       76001., 54001.,  5001., 11001., 41001., 76001.,  5001., 13001.,
       11001., 41524., 11001., 11001., 25214., 15001., 11001., 20001.,
        5001., 76001., 11001., 66001., 11001., 11001., 13001., 17877.,
       11001., 13001., 11001., 11001., 11001.,  5001.,  5001.,  5001.,
       41001., 66001.,  5001., 11001., 13001.,  5001., 76001., 41001.,
       11001., 11001., 11001.,  5001., 76001., 11001., 68276., 11001.,
       76520., 66001., 76126., 11001., 76001., 11001., 11001., 11001.,
       76001., 41001., 11001., 20001., 11001., 11001., 20001., 20001.,
       11001., 11001.,  5467., 76001.,  8001., 76001., 11001., 11001.,
       11001., 76001.,  5001., 76001., 11001.,  5001., 68001., 76001.,
       68001.,  5001., 11001., 11001.,  8001., 20001., 11001., 11001.,
       76001., 11001., 11001.,  5001., 13001., 76001., 17877., 11001.,
       11001.,  5001., 11001., 11001., 11001., 25214., 63001., 11001.,
       11001., 68001., 54001.,  8001., 54001., 11001., 11001., 11001.,
       68276., 66001., 11001., 41001., 11001., 19001., 17380., 76520.,
       13001., 11001., 66001., 11001., 11001., 76001., 68276., 76001.,
       66001., 11001., 76001., 11001., 11001.,  5001., 25754., 19001.,
       11001., 11001., 76001., 11001., 11001., 63001., 11001., 88001.,
       76001.,  5120., 11001., 11001.], dtype=float32)
```

GaussianNB implementa el algoritmo *Gaussian Naive Bayes* para la clasificación. Se supone que la probabilidad de las características es gaussiana:

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

Usamos la **matriz de confusión** y la **puntuación de precisión** comparando los valores de prueba predichos y reales:

```
cm = confusion_matrix(y_test, y_pred)
ac = accuracy_score(y_test,y_pred)
cm
```

```
array([[16,  0,  0, ...,  0,  0,  0],
       [ 1,  0,  0, ...,  0,  0,  0],
       [ 2,  0,  0, ...,  0,  0,  0],
       ...,
       [ 0,  0,  0, ...,  0,  0,  0],
       [ 0,  0,  0, ...,  0,  0,  1],
       [ 0,  0,  0, ...,  0,  0,  0]])
```

```
[34] ac
```

```
0.7666666666666667
```

Se obtuvo un buen porcentaje del 76% con este modelo ♥

8. Contratiempos.

Se encontraron pocas variables numéricas en el set. Además, al ser bajados como objetos algunas funciones arrojaron problemas en la conversión, por ejemplo la función .train(). Sin embargo, el problema fue resuelto satisfactoriamente.

9. Conclusiones.

- Un análisis comparativo entre la tasa de propagación y las edades de las personas sin importar su densidad poblacional de las áreas en estudio demuestra qué en su mayoría es en la población joven en quién recae la mayor tasa de propagación del virus. Estos resultados en principio podrían corroborar la hipótesis qué estipula qué son las poblaciones más jóvenes las que poseen menos conciencia colectiva sobre el riesgo de la enfermedad. Está conclusión podría validar estudios recientes qué sugieren que durante la pandemia fueron las poblaciones más jóvenes las menos responsables ante la autoridad sanitaria y que por ende es en ellos donde se encuentra una proliferación más rápida del virus.

- Con respecto a los modelos de machine learning podemos concluir que el modelo de **regresión lineal** demostró no ser el indicado para manejar nuestros datos pues arrojó un score incoherente. En su lugar, se pudo predecir correctamente la variable "ciudad_municipio" en funsión del "departamento" y la "edad" con el modelo **Naive bayes**. Por ultimo, pudimos predecir la variable categorica "sexo" en funsión del "departamento" y la "edad" usando un modelo de regresion logistico.

10. Referencias

- Coronavirus Colombia. (n.d.). Coronavirus Colombia. Retrieved December 2, 2021, from https://www.ins.gov.co/Noticias/paginas/coronavirus.aspx

- Presidencia de la republica. (n.d.). Presidencia de La Republica. Retrieved December 2, 2021, from https://coronaviruscolombia.gov.co/Covid19/index.html

- https://infogram.com/panorama-general-1h7z2lgn3l9l4ow?live

- https://www.hosteltur.com/126132_tendencias-de-los-viajes-en-2019-por-edad-y-nacionalidad.html